

Το πρόβλημα της μέγιστης υποακολουθίας

Κώστας Άγγελος

2 Νοεμβρίου 2022

Περίληψη

Με τον όρο πολυπλοκότητα ορίζουμε την διαδικασία υπολογισμού ενός τύπου ο οποίος προσδιορίζει τον συνολικό χρόνο που απαιτείται για την εκτέλεση ενός συγκεκριμένου αλγορίθμου για είσοδο δεδομένων μεγέθους (n). Στην συγκεκριμένη εργασία αναλύονται τρεις αλγόριθμοι που λύνουν το πρόβλημα της μέγιστης υποακολουθίας. Για κάθε έναν ξεχωριστά υπολογίζουμε την πολυπλοκότητα, τονίζοντας έτσι την διαφορά σε ταχύτητα λόγω της επιλογής ενός ικανότερου αλγορίθμου.

1. Εισαγωγή

Το πρόβλημα ουσιαστικά είναι το εξής : Δεδομένου μια ακολουθία ακεραίων τιμών μας ζητείται να βρούμε την υποακολουθία με το μεγαλύτερο άθροισμα. Στην συγκεκριμένη αναφορά παρουσιάζονται τρεις αλγόριθμοι που λύνουν το πρόβλημα :

1) Brute Force Algorithm

Αποτελεί τον πιο απλό τρόπο υπολογισμού της υποακολουθίας γι' αυτό και έχει πολυπλοκότητα ίση με $O(n^3)$.

2) Prefix Algorithm

Αποτελεί μια καλύτερη προσέγγιση με πολυπλοκότητα ίση με $O(n^2)$.

3) Kadane Algorithm

Αποτελεί τον πιο γρήγορο τρόπο επίλυσης του προβλήματος με πολυπλοκότητα ίση με $O(n)$.

2. Πειραματικά αποτελέσματα

Για την καταγραφή των πειραματικών αποτελεσμάτων χρησιμοποιήθηκε το εξής setup :

- IDE : PyCharm 2022.2.3
- Γλώσσα προγραμματισμού : Python 3.10.8
- Βιβλιοθήκη γραφικών : matplotlib

Χαρακτηριστικά υπολογιστή :

- CPU : AMD Ryzen 5 3400g
- RAM : 32 GB DDR4 (2933 MHz)

Παράμετροι πειραμάτων :

Για την μέτρηση του χρόνου εκτέλεσης των αλγορίθμων χρησιμοποιήθηκαν τυχαίοι αριθμοί με εύρος το $[-100, 100]$. Για κάθε έναν αλγόριθμο χρησιμοποιήθηκε πλήθος από διάφορα N . Στην συγκεκριμένη αναφορά έχουμε: $N \in \{100, 500, 1.000, 5.000\}$.

Παρακάτω παρουσιάζονται τα αποτελέσματα του χρόνου εκτέλεσης (σε Seconds) για διαφορετικό αριθμό από N .

Ένα sample της εξόδου του προγράμματος :

```
Brute Force Approach --> max is: 197      start index is : 42      end index is : 51      time elapsed : 0.009945 seconds
Brute Force Approach --> max is: 2328     start index is : 8       end index is : 490     time elapsed : 1.081398 seconds
Brute Force Approach --> max is: 1611     start index is : 744     end index is : 946     time elapsed : 9.123631 seconds
Brute Force Approach --> max is: 2951     start index is : 799     end index is : 1304    time elapsed : 1234.102419 seconds

Prefix Approach --> max is: 197          start index is : 42      end index is : 51      time elapsed : 0.0005 seconds
Prefix Approach --> max is: 2328         start index is : 8       end index is : 490     time elapsed : 0.014298 seconds
Prefix Approach --> max is: 1611         start index is : 744     end index is : 946     time elapsed : 0.061552 seconds
Prefix Approach --> max is: 2951         start index is : 799     end index is : 1304    time elapsed : 1.438411 seconds

Kadane Algorithm Approach --> max is: 197      start index is : 42      end index is : 51      time elapsed : 4.6e-05 seconds
Kadane Algorithm Approach --> max is: 2328     start index is : 8       end index is : 490     time elapsed : 0.000204 seconds
Kadane Algorithm Approach --> max is: 1611     start index is : 744     end index is : 946     time elapsed : 0.000351 seconds
Kadane Algorithm Approach --> max is: 2951     start index is : 799     end index is : 1304    time elapsed : 0.001682 seconds
```

Εικόνα 1. Output κυρίως προγράμματος

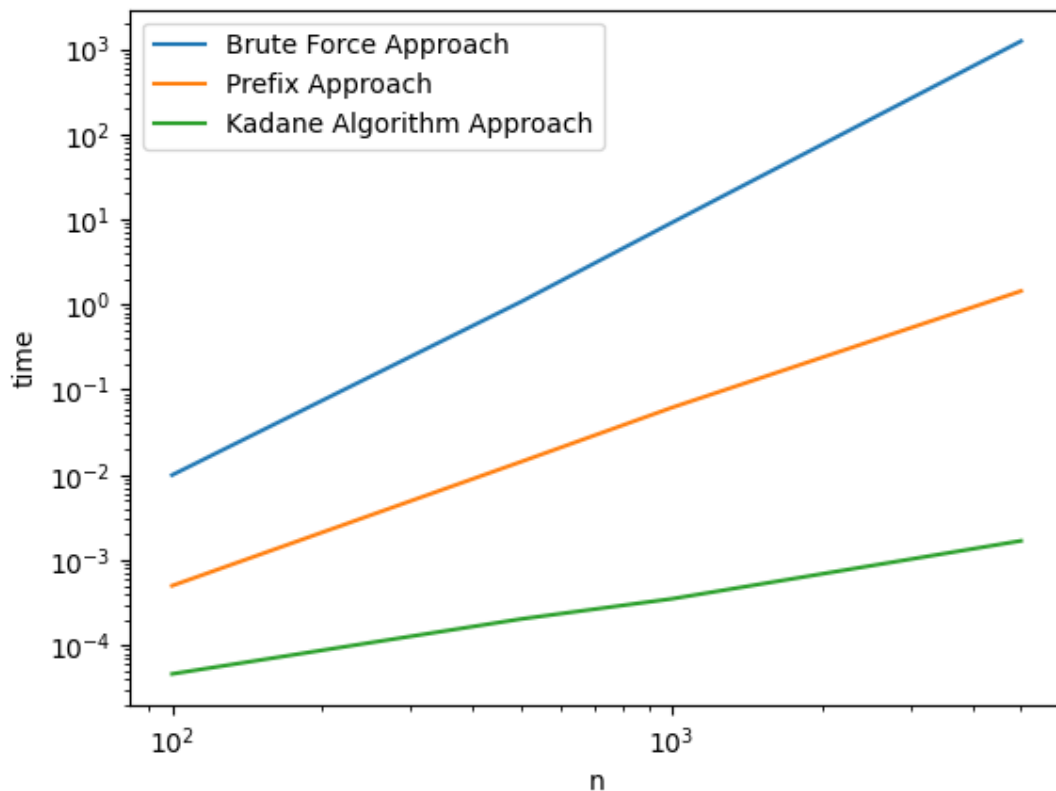
```
PS C:\Users\Leader\Desktop\alco_assignment_1> py tests.py
...
-----
Ran 3 tests in 0.000s

OK
```

Εικόνα 2. Output των tests

	N=100	N=500	N=1.000	N=5.000
Brute Force Algorithm	0.009945	1.081398	9.123631	1234.102419
Prefix Algorithm	0.0005	0.014298	0.061552	1.438411
Kadane Algorithm	4.6e-05	0.000204	0.000351	0.001682

Πίνακας 1. Πίνακας αποτελεσμάτων



Εικόνα 3. Οπτικοποίηση αποτελεσμάτων (matplotlib)

2. Συμπεράσματα

Από το συγκεκριμένο πρόβλημα αλλά και τις μετρήσεις που εκτελέσαμε φαίνεται η χρησιμότητα σε χρόνο και χώρο από την επιλογή ενός ικανού και γρηγορότερου αλγόριθμου σε σχέση με απλούστερες υλοποιήσεις. Συγκεκριμένα υπάρχει τεράστια διαφορά στο performance ιδίως εάν χρησιμοποιήσουμε μεγάλο αριθμό σε N. Για παράδειγμα για N=10.000 ο brute force μπορεί να διαρκέσει έως και 3.5 ώρες μέχρι να βρει λύση για το πρόβλημα (ανάλογα πάντα και με τα χαρακτηριστικά στα οποία διεξάγεται το πείραμα). Για την συγκεκριμένη αναφορά το εύρος του N επιλέχθηκε με γνώμονα να υπάρχει ισορροπία μεταξύ ταχύτητας και efficiency.